

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение  
высшего образования  
«Самарский национальный исследовательский университет имени  
академика С. П. Королева»  
(Самарский университет)

Институт информатики и кибернетики  
Прикладные математика и физика

**Отчёт по курсовой работе**

Дисциплина: “Технологии программирования”

Преподаватель: Белоусов А. А.

---

(подпись)

Студент: Чернов В. А.

6301-030301D

---

(подпись)

Самара 2023

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ .....	3
ВВЕДЕНИЕ.....	4
1 ПЛАН ПРОЕКТА .....	5
2 СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ С БИЗНЕС-ЦЕЛЯМИ, БИЗНЕС-ЗАДАЧАМИ И ПОЛЬЗОВАТЕЛЬСКИМИ СЦЕНАРИЯМИ .....	6
3 ОПИСАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ .....	9
4 ИНСТРУКЦИЯ ПО РАЗВЁРТЫВАНИЮ И УСТАНОВКЕ СИСТЕМЫ.....	10
5 ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ СИСТЕМЫ .....	17
ЗАКЛЮЧЕНИЕ .....	24
ПРИЛОЖЕНИЕ .....	25

## **ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ**

Разработать проект информационной системы и реализовать разработанный проект в соответствии с технологией разработки ПО.

## **ВВЕДЕНИЕ**

В данной курсовой работе в качестве предметной области было выбрано веб-приложение. Суть данного проекта заключается в использовании технологий машинного обучения с целью классификации каких-либо данных, которые будет выполнять классификатор, и, в следствие, предсказывания дальнейших событий.

**Цель:** создание веб-приложения, использующего технологии машинного обучения.

Необходимо разработать веб-приложение выполняющее следующие задачи:

- Предсказание дальнейших значений на основе входных данных;

Веб-приложение должно соответствовать следующим требованиям:

- Иметь интуитивно понятное меню;
- Иметь панель входных данных;

### **Описание:**

Данный проект реализуется в рамках упрощения задачи расчёта и разведочного анализа данных. Веб-приложение старается облегчить работу пользователя по классификации и анализу некоторых данных.

# 1 ПЛАН ПРОЕКТА

План работ по проекту с основами вехами и датами их выполнения:

1. Подготовка описания проекта и его основных задач - 06.11.2023;
2. Сценарии использования системы с бизнес-целями, бизнес-задачами и пользовательскими сценариями - 15.11.2023;
3. Описание архитектуры системы - 22.11.2023;
4. Код реализации системы - 30.11.2023;
5. Инструкции по развертыванию и установке системы - 04.12.2023;
6. Инструкция пользователя системы - 11.12.2023;
7. Оформление отчёта - 14.12.2023.

Этапы проекта отслеживаются в диаграмме Ганта с учётом всех основных вех и дат их исполнения (рисунок 1).

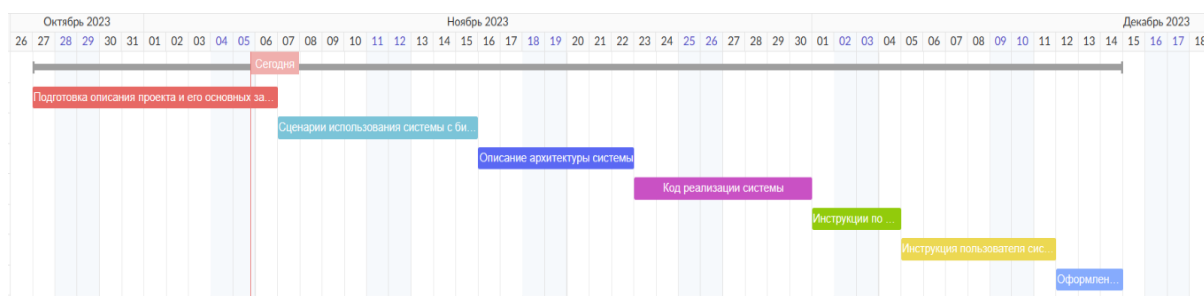


Рисунок 1

## **2 СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ С БИЗНЕС-ЦЕЛЯМИ, БИЗНЕС-ЗАДАЧАМИ И ПОЛЬЗОВАТЕЛЬСКИМИ СЦЕНАРИЯМИ**

Веб-приложение - сайт с моделью, использующей технологии машинного обучения, которая будет классифицировать данные, а также прогнозировать дальнейшие результаты.

Данный раздел описывает связь функций системы с бизнес-целями и сценариями использования.

### **Роли пользователей:**

- Студенты;
- Научные сотрудники;
- Лаборанты;
- Программисты.

### **Бизнес-цели:**

**BG1    Увеличение числа пользователей сайта через предоставление качественного сервиса классификации на основе методов машинного обучения**

**F1-1    Предоставление различного функционала**

**UC1-1-1    Выбор данных**

Я, как пользователь, хочу загружать свои данные для их классификации

<b>Актор</b>	<b>Пользователь</b>
<b>Триггер</b>	Пользователь открыл боковую панель

<b>Результат</b>	Просмотр бегунков для входных данных
<b>Основной поток</b>	<ol style="list-style-type: none"> <li>1. Пользователь находится на главной странице</li> <li>2. Пользователь открывает боковую панель</li> <li>3. Перед пользователем входные данные</li> <li>4. Пользователь загружает свой набор входных данных</li> </ol>

### UC1-1-2 Выбор модели классификации

Я, как пользователь, хочу выбирать модель классификации данных

<b>Актор</b>	<b>Пользователь</b>
<b>Триггер</b>	Пользователь открыл боковую панель
<b>Результат</b>	Просмотр раздела “модель классификации”
<b>Основной поток</b>	<ol style="list-style-type: none"> <li>1. Пользователь находится на главной странице</li> <li>2. Пользователь открывает боковую панель</li> <li>3. Перед пользователем выбор модели</li> </ol>

### UC1-1-3 Проверки модели классификации на устойчивость

Я, как пользователь, хочу знать, устойчива ли модель, которую выбрал

<b>Актор</b>	<b>Пользователь</b>
<b>Триггер</b>	Пользователь получил некоторые данные

<b>Результат</b>	Просмотр раздела “устойчивость данной модели”
<b>Основной поток</b>	<ol style="list-style-type: none"> <li>1. Пользователь находится на главной странице</li> <li>2. Пользователь получает некоторые данные после работы классификатора</li> <li>3. Перед пользователем устойчивость модели классификации, которую он выбрал</li> </ol>



### 3 ОПИСАНИЕ АРХИТЕКТУРЫ СИСТЕМЫ

Данный раздел описывает необходимые сервисы, приложения и технологии для создания курсового проекта.

#### Написание и запуск веб-приложения при помощи Python

Чтобы построить модель и опубликовать её где-нибудь, понадобятся библиотеки streamlit, pandas и scikit-learn. Взглянем на общую схему проекта. Он будет состоять из двух больших частей: Frontend и Backend.

Во Frontend-части приложения, а именно, на веб-странице, будет боковая панель, находящаяся слева, в которой можно будет вводить входные параметры модели. Эти данные будут передаваться Backend, где предварительно обученная модель будет производить классификацию, используя заданные характеристики. Результаты классификации отправляются Frontend.

В Backend-части приложения то, что ввёл пользователь, сохраняется в датафрейме, который будет использоваться в виде тестовых данных для модели. Потом будет построена модель для обработки данных. В ней будет применяться алгоритм из библиотеки scikit-learn. И наконец, модель будет применена для классификации данных, введенных пользователем. Кроме того, будут возвращаться и данные о прогностической вероятности. Это позволит нам определить степень достоверности результатов классификации.

# 4 ИНСТРУКЦИЯ ПО РАЗВЁРТЫВАНИЮ И УСТАНОВКЕ СИСТЕМЫ

## 1. Подготовка к установке

### 1.1. Установка python 3.8.0

Скачайте официальные установочные файлы с официального сайта <https://code.visualstudio.com/Download> версии 3.8.0, либо же последней версии. Данный редактор является кроссплатформенным, поэтому его можно установить на операционные системы Windows, Linux или же MAC. Также устанавливаем интерпретатор Python <https://www.python.org> с официального сайта.

### 1.2. Установка расширений

Кликаем по значку “Расширения” или же используем комбинацию клавиш CTRL+SHIFT+X.

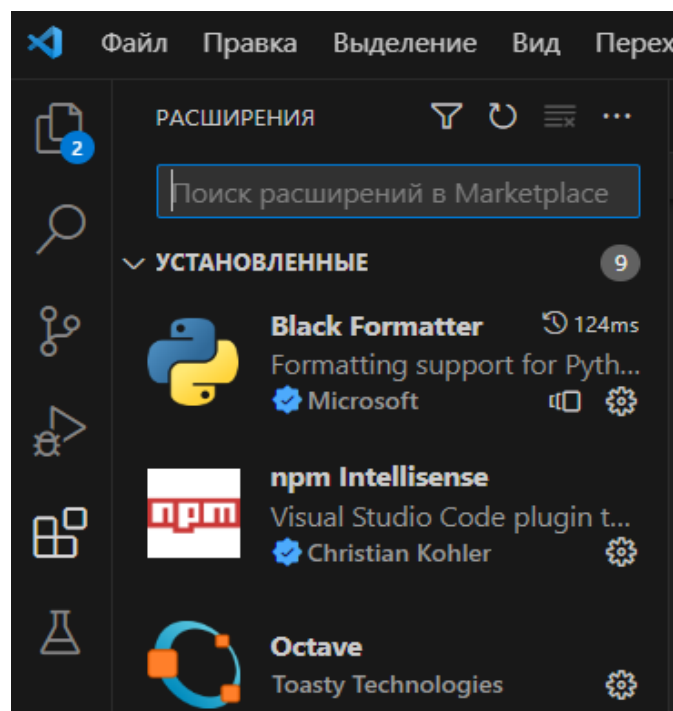


Рисунок 1.1 – Инструкция по скачиванию файла

В появившемся окне пишем: “Russian Language Pack” - это установка русского языка.

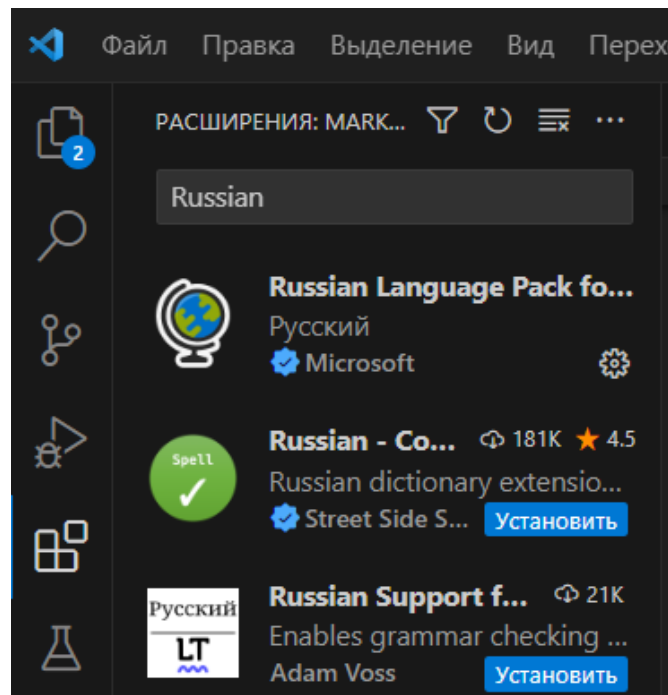


Рисунок 1.2 – Инструкция по скачиванию файла

Устанавливаем данное расширение и перезапускаем VS Code.

Установим сам Python. Так же заходим в “Расширения” и в появившемся окне вводим “Python”.

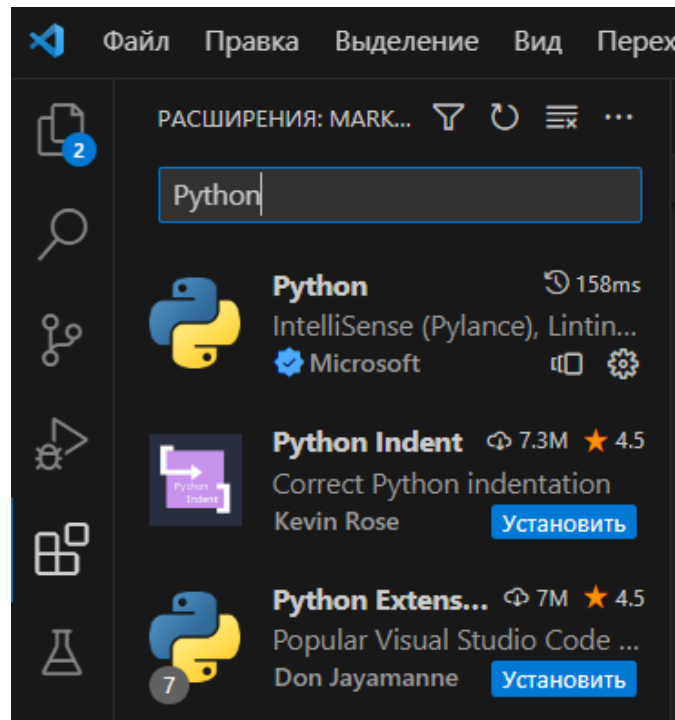


Рисунок 1.3 – Инструкция по скачиванию файла

Устанавливаем первое расширение.

Далее нужно выбрать интерпретатор Python. Для этого нажимаем комбинацию клавиш CTRL+SHIFT+P и в появившемся окне пишем “select interpreter”.

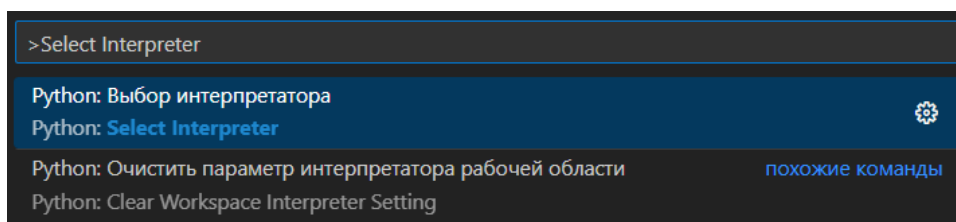


Рисунок 1.4 – Инструкция по скачиванию файла

Выбираем интерпретатор и вводим путь, где у вас установлен интерпретатор Python.

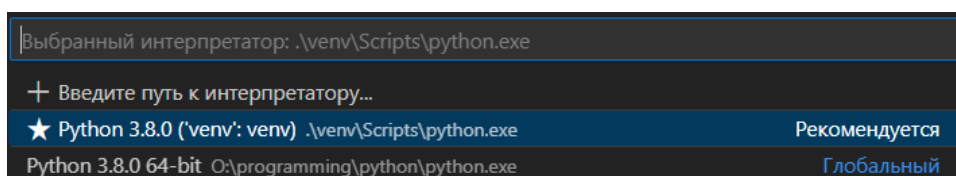


Рисунок 1.5 – Инструкция по скачиванию файла

Всё готово для работы. Пробуем создать новый файл, для этого в левом верхнем углу нажимаем “Файл” > “Сохранить” и сохраняем его с расширением “.py”.

## 2. Установка веб-приложения

2.1 Перейти в репозиторий GitHub: [SHPATELb/- \(github.com\)](https://github.com/SHPATELb/-)

2.2 Скачать файл u.py

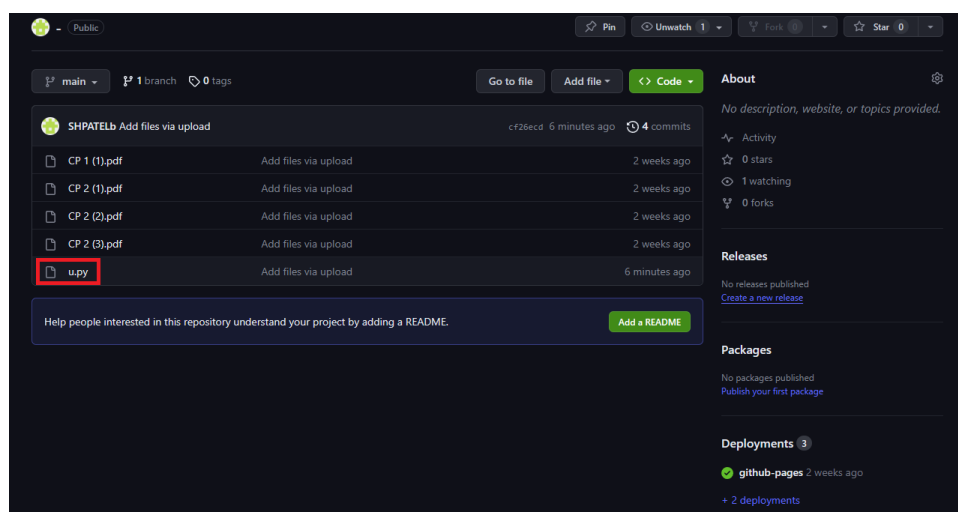


Рисунок 2.1 – Инструкция по скачиванию файла

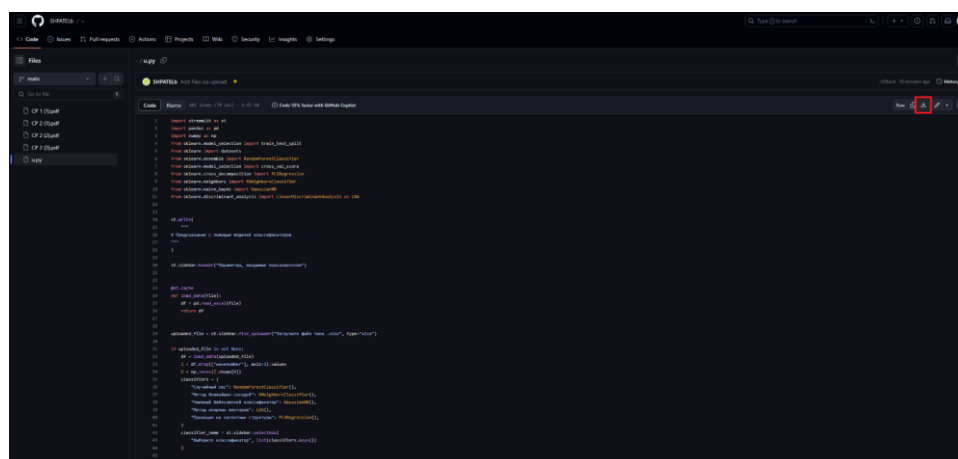


Рисунок 2.2 – Инструкция по скачиванию файла

## 3. Открытие скаченного файла

Перейдите в дерикторию, куда был установлен файл и откройте его с помощью VS Code.

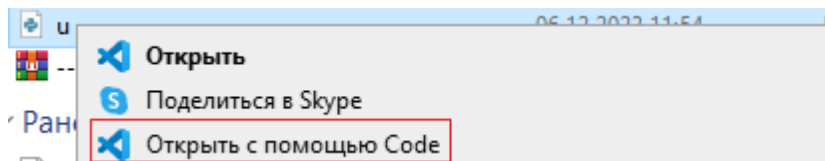


Рисунок 3.1 – Инструкция по открытию файла

#### 4. Установка нужных библиотек

Для работы данного веб-приложения нужно установить следующие библиотеки:

- Streamlit;
- Pandas;
- Numpy;
- Scikit-learn;
- Openpyxl.

Установку нужно производить в терминале при помощи команды `pip install`:

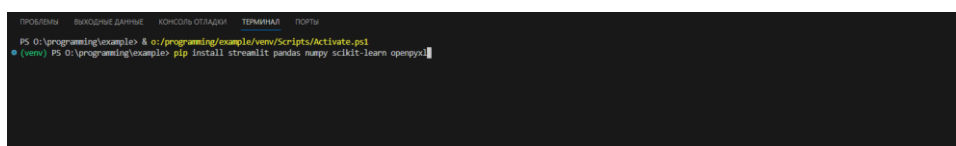


Рисунок 4.1 – Инструкция по установке библиотек

#### 5. Запуск программы

В правом верхнем углу редактора нажимаем кнопку запуска.

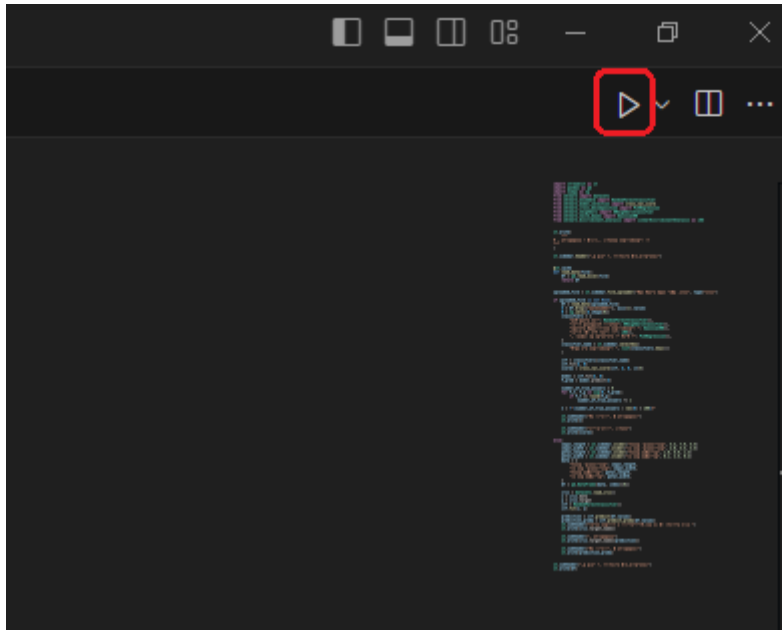


Рисунок 5.1 – Инструкция по запуску программы

После запуска программы в терминале появится что-то подобное:

```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ
PS 0:\programming\example> & o:/programming/example/venv/Scripts/Activate.ps1
• (venv) PS 0:\programming\example> & o:/programming/example/venv/Scripts/python.exe o:/programming/example/u.py
2023-12-06 12:53:40.911
Warning: to view this Streamlit app on a browser, run it with the following
command:
streamlit run o:/programming/example/u.py [ARGUMENTS]
• (venv) PS 0:\programming\example>
```

Рисунок 5.2 – Инструкция по запуску программы

В нижней строке нужно прописать команду: `streamlit run u.py`. После чего откроется окно в браузере.

```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ

PS O:\programming\example> & o:/programming/example/venv/Scripts/Activate.ps1
• (venv) PS O:\programming\example> & o:/programming/example/venv/Scripts/python.exe o:/programming/example/u.py
2023-12-06 12:53:40.911
Warning: to view this Streamlit app on a browser, run it with the following
command:

    streamlit run o:/programming/example/u.py [ARGUMENTS]
• (venv) PS O:\programming\example> streamlit run u.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.0.105:8501

A new version of Streamlit is available.

See what's new at https://discuss.streamlit.io/c/announcements

Enter the following command to upgrade:
$ pip install streamlit --upgrade
```

Рисунок 5.3 – Инструкция по запуску программы

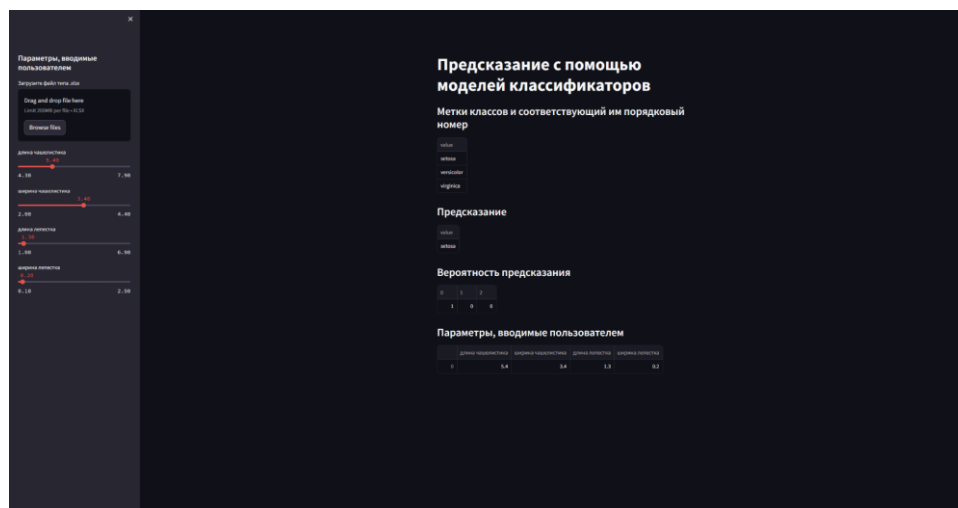


Рисунок 5.4 – Инструкция по запуску программы



## **5 ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ СИСТЕМЫ**

Данная инструкция содержит область и назначение применения, а также описание элементов интерфейса пользователя и описание базовых сценариев использования системы в виде примеров со скриншотами.

### **1. Введение**

Суть данного проекта заключается в использовании технологий машинного обучения с целью классификации каких-либо данных, которые будет выполнять классификатор, и, в следствии, предсказывания дальнейших событий.

#### **1.1. Область применения**

Данный сайт реализуется в рамках упрощения задачи расчёта и разведочного анализа данных. Веб-приложение старается облегчить работу пользователя по классификации и анализу некоторых данных.

#### **1.2. WEB-интерфейс**

Сайт состоит из главного меню и боковой панели.

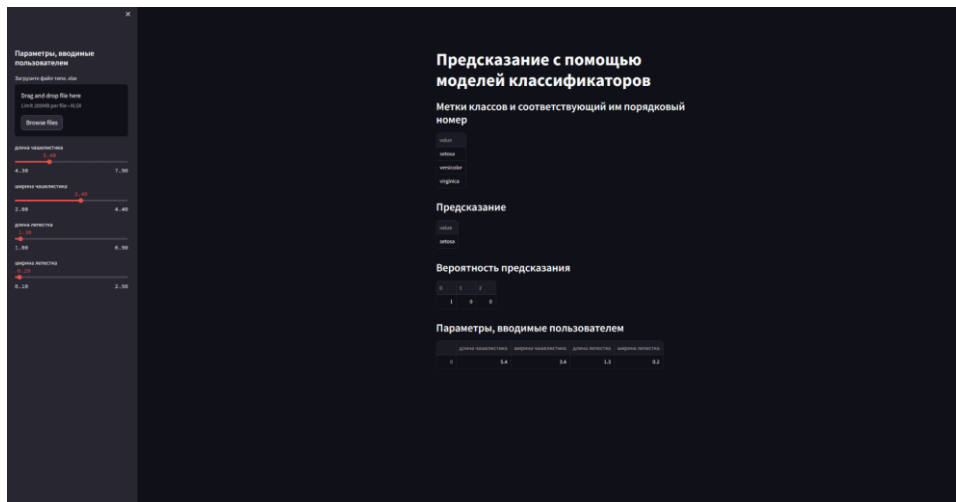


Рисунок 1.1 – Инструкция по запуску программы

Главное меню представляет из себя информационную страницу, на которую выводятся все полученные значения в результате классификации. Боковая панель отображает входные данные.

## 2. Назначения и условия применения

### 2.1. Цель веб-приложения

- Предсказание дальнейших значений на основе входных данных.

### 2.2. Условия применения системы

Для корректной работы программы нужно версия Python 3.8.0 или же выше.

Также нужны библиотеки streamlit, pandas, numpy, scikit-learn и openruhl.

## 3. Работа с сервисом

### 3.1. Подготовка системы к работе

Для работы с сайтом Вам необходимо открыть документ «Инструкция по развёртыванию и установке системы» и выполнить все написанные в нём пункты.

### 3.2. Описание работы с сервисом

Пока вы не загрузили какой-либо файл формата .xlsx для классификации своих данных, вы можете изначально попробовать модель, определяющую вид цветка ириса.

Изменяя параметры на боковой панели, а именно: “длина чашелистика”, “ширина чашелистика”, “длина лепестка”, “ширина лепестка” - вы можете увидеть к какому виду относится цветок ириса с данными параметрами в графе: “Предсказание”.

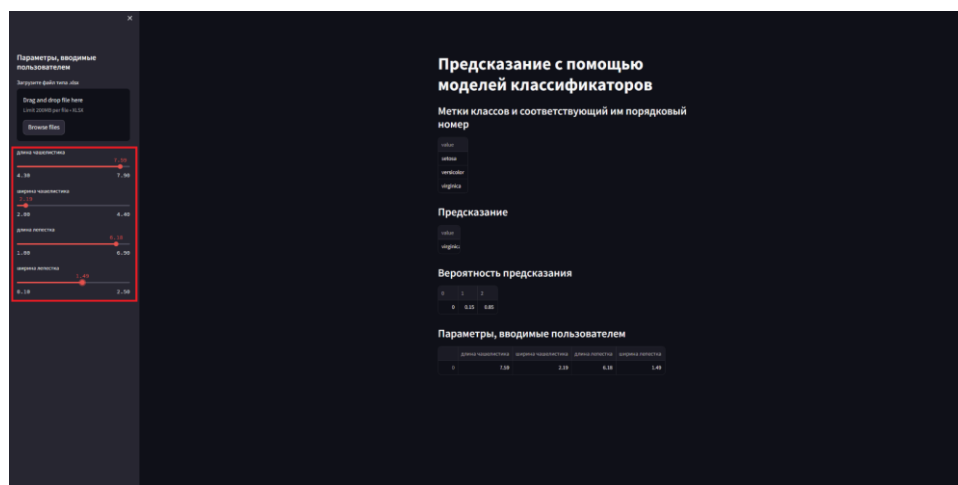


Рисунок 3.1 – Параметры боковой панели

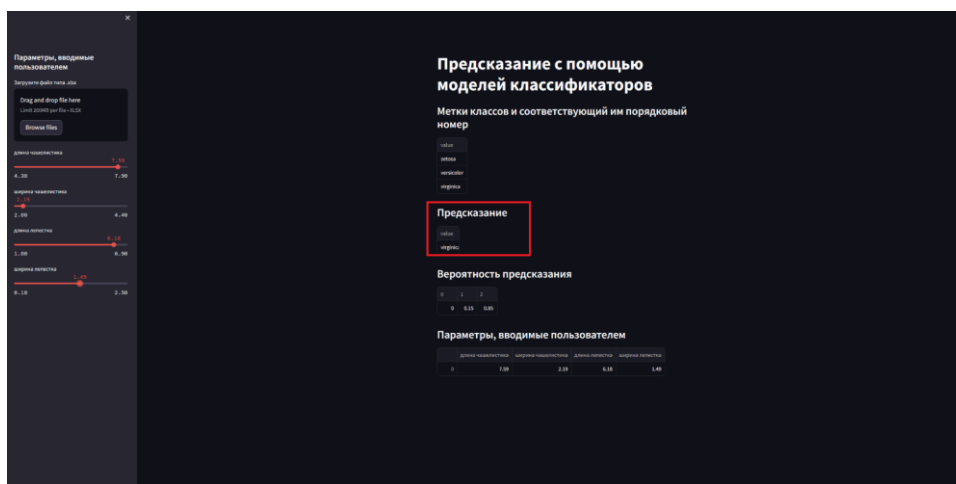


Рисунок 3.2 – Графа “Предсказание”

В графе “Вероятность предсказания” вы можете понаблюдать, в каком процентном соотношении вид цветка соответствует данной разновидности с заданными параметрами.

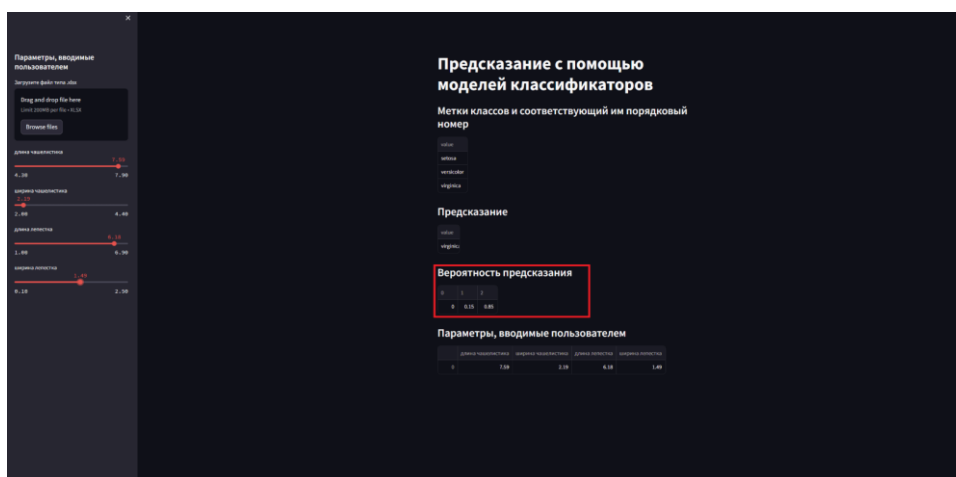


Рисунок 3.3 – Графа “Вероятность предсказания”

Пользователь может загружать свои данные в виде таблицы формата .xlsx в графе “Параметры, вводимые пользователем”, находящейся на боковой панели. Нужно нажать кнопку “Browse files”, и выбрать загружаемый файл.

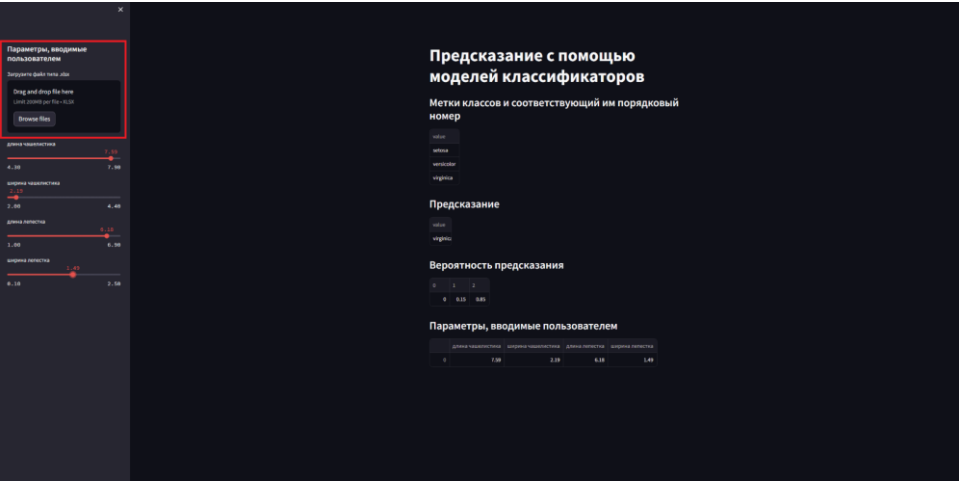


Рисунок 3.4 – Графа “Параметры, вводимые пользователем”

После загрузки ваших данных в виде таблице формата .xlsx, вы перейдёте на следующую страницу.

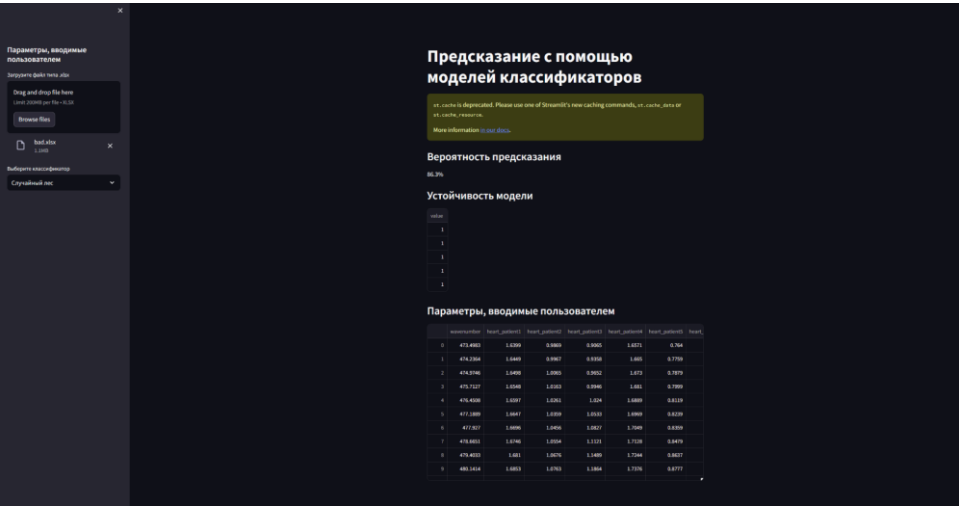


Рисунок 3.5 – Меню с данными пользователя

На боковой панели можно выбрать одну из пяти моделей классификации данных.

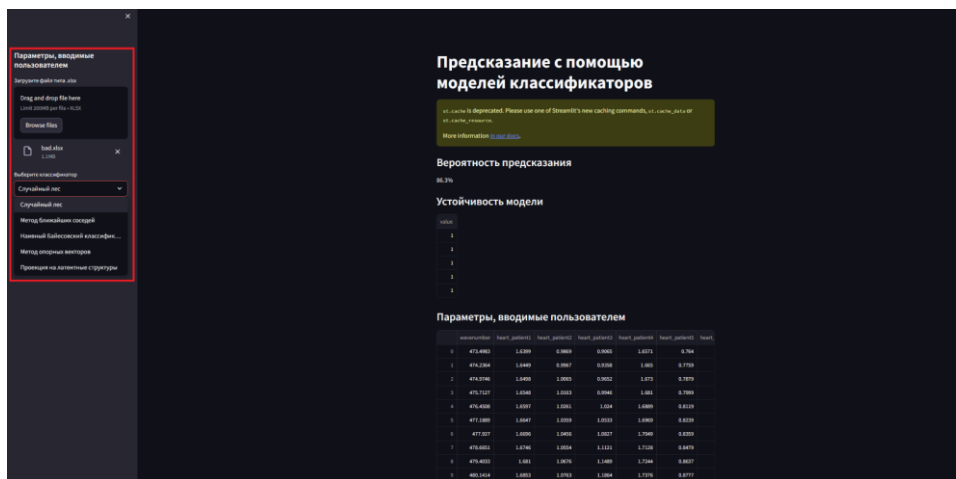


Рисунок 3.6 – Выбор модели классификации данных

В главном меню есть графы “Вероятность предсказания” и “Устойчивость модели”.

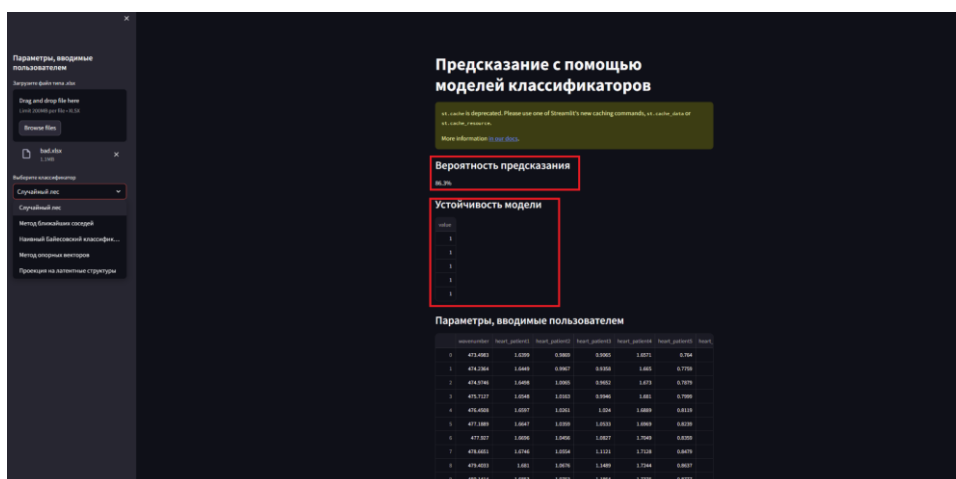


Рисунок 3.7 – Графы “Вероятность предсказания” и “Устойчивость модели”

По вероятности предсказания мы можем узнать насколько хорошо подходит классификатор для предоставленных данных, и какой подход лучше всего использовать для их же анализа.

Устойчивость модели показывает нам, как ведёт себя классификатор при перемешивании данных.

## 4. Аварийные ситуации

В случае отказа или сбое работы системы, а также для получения консультаций и технической поддержки необходимо обратиться по электронной почте: [vsevolod.67@yandex.ru](mailto:vsevolod.67@yandex.ru).

## **ЗАКЛЮЧЕНИЕ**

В ходе курсовой работы было разработано веб-приложение, использующее технологии машинного обучения. Пользователь может классифицировать и анализировать необходимые ему данные.

В качестве среды разработки был выбран Visual Studio Code. В качестве языка программирования был выбран Python. Для создания веб-приложения была выбрана библиотека streamlit.



## **ПРИЛОЖЕНИЕ**

Код разработанного проекта находится в github:  
<https://github.com/SHRATeLb/>